

**Method and circuit for decoding convolutional codes**

Patent Number: ☐ US5327441  
Publication date: 1994-07-05  
Inventor(s): KAWAZOE KATSUHIKO (JP); HONDA SHUNJI (JP); KATO SHUZO (JP); KUBOTA SHUJI (JP)  
Applicant(s): NIPPON TELEGRAPH & TELEPHONE (JP)  
Requested Patent: ☐ JP5244019  
Application Number: US19920991215 19921215  
Priority Number(s): JP19910331640 19911216  
IPC Classification: G06F11/10  
EC Classification: H03M13/23, H03M13/41  
Equivalents:

---

**Abstract**

---

In a simple decoder which decodes convolutional codes of constraint length  $K$  and coding rate  $n/m$ , encoded data which is supplied thereto in steps of  $m$  bits are distributed, bit by bit, to  $m$  shift registers each having  $x = [(K-1)/(m-n)]$  series-connected shift stages. Here,  $[p]$  means the minimum integer equal to or larger than a real number  $p$ . The connection of  $n$  modulo-2 addition circuits to all the shift stages of all the shift registers is defined by  $n$  decoding generative vectors which define  $n$  decoding generative polynomials. The modulo-2 addition circuits perform modulo-2 additions of the outputs of the shift stages connected thereto and output  $n$  results of additions as decoded results of  $n$  bits. The  $n$  decoding generative vectors are selected from decoding generative vectors of  $N$  rows which are obtained as an inverse matrix of a square matrix whose elements are  $N \times N$  coefficients which define  $N = m \times$  convolutional code generating polynomials.

---

Data supplied from the esp@cenet database - I2

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-244019

(43)公開日 平成5年(1993)9月21日

(51)Int.Cl.<sup>5</sup>

H 0 3 M 13/12

H 0 4 L 25/08

識別記号

庁内整理番号

7259-5 J

B 8226-5K

F I

技術表示箇所

審査請求 未請求 請求項の数12(全 18 頁)

(21)出願番号 特願平4-336292

(22)出願日 平成4年(1992)12月16日

(31)優先権主張番号 特願平3-331640

(32)優先日 平3(1991)12月16日

(33)優先権主張国 日本(J P)

特許法第30条第1項適用申請有り 1991年8月15日 社団法人電子情報通信学会発行の「1991年電子情報通信学会秋季大会講演論文集」に発表

(71)出願人 000004226

日本電信電話株式会社

東京都千代田区内幸町一丁目1番6号

(72)発明者 川添 雄彦

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(72)発明者 本田 俊二

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(72)発明者 久保田 周治

東京都千代田区内幸町1丁目1番6号 日

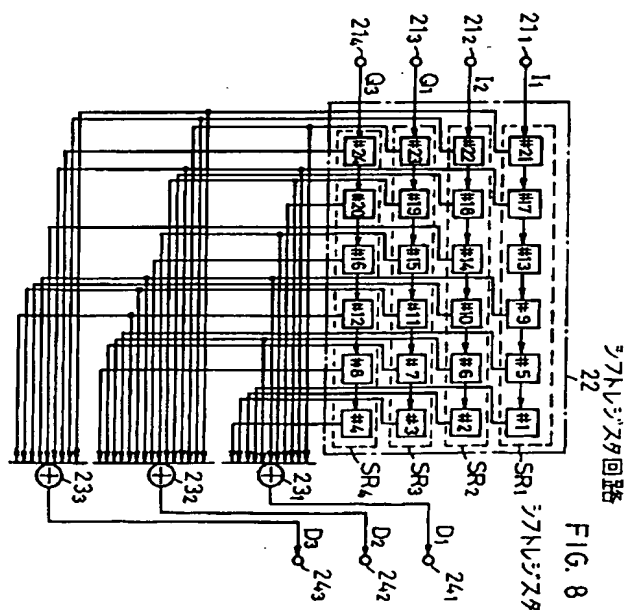
本電信電話株式会社内

(74)代理人 弁理士 草野 卓

最終頁に続く

(54)【発明の名称】 畳み込み符号の簡易復号方法及び簡易復号回路

(57)【要約】

【目的】 所望の拘束長K、符号化率 $n/m$ の畳み込み符号に対する簡易復号を行う。【構成】 復号回路に供給される $m (=4)$ ビット毎の畳み込み符号データは、それぞれが $x = [(K-1)/(m-n)]$ 個の直列接続されたシフト段を有する $m$ 個のシフトレジスタ $SR_1 \sim SR_m$ に1ビットずつ分配されて与えられる。ここで $[p]$ は実数 $p$ 以上の最小の整数を意味する。 $n (=3)$ 個の $\text{mod } 2$ 加算器(23<sub>1</sub>、23<sub>2</sub>、23<sub>3</sub>)はそれぞれ全シフトレジスタの全シフト段(#1~#24)と $n$ 個の復号生成多項式を規定する $n$ 個の復号生成ベクトルによって結合が規定され、それぞれの $\text{mod } 2$ 加算器は結合されたシフト段の出力を $\text{mod } 2$ 加算し、 $n$ 個の加算結果を $n$ ビットの復号結果として出力する。 $n$ 個の復号生成ベクトルは、 $N=mx (=24)$ 個の畳み込み符号の生成多項式を規定する $N \times N$ 個の係数を要素とする正方行列の逆行列として得られる $N$ 行の復号生成ベクトルから選ばれた連続する $n$ 行の要素をそれぞれ係数として持つものである。

## 【特許請求の範囲】

【請求項 1】 符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号回路であり、 $K$ 、 $n$ 、 $m$  は自然数、 $K>1$ 、 $m>n$  であり、

順次受信した  $m$  ビット毎の上記畳み込み符号のデータがそれぞれ 1 ビットずつ供給され、それぞれが  $x \geq (K-1)/(m-n)$  を満足する予め決めた整数  $x$  個の直列接続されたシフト段を有する  $m$  個のシフトレジスタと、上記  $m$  個のシフトレジスタの全  $N=mx$  個のシフト段は  $N$  個の符号データを保持するためのシフトレジスタ回路を構成し、

上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  個のシフト段から選択したそれぞれ複数の上記シフト段から出力を取り出す  $n$  個の選択結合手段と、

上記  $n$  個の選択結合手段とそれぞれ接続され、それぞれ上記複数のシフト段の出力の  $\text{mod } 2$  加算を行い  $n$  ビットの復号データを出力する  $n$  個の  $\text{mod } 2$  加算手段、とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続する  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N=mx$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である。

【請求項 2】 符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号回路であり、 $K$ 、 $n$ 、 $m$  は自然数、 $K>1$ 、 $m>n$  であり、

受信した上記畳み込み符号のデータが 1 ビットずつ供給され、 $N \geq m(K-1)/(m-n)$  を満足する予め決めた整数  $N$  個の直列接続されたシフト段を有するシフトレジスタと、上記  $N$  個のシフト段は  $N$  個の符号データを保持するためのシフトレジスタ回路を構成し、

上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  個のシフト段から選

択したそれぞれ複数の上記シフト段から出力を取り出す  $n$  個の選択結合手段と、

上記  $n$  個の選択結合手段とそれぞれ接続され、それぞれ上記複数のシフト段の出力の  $\text{mod } 2$  加算を行い  $n$  ビットの復号データを出力する  $n$  個の  $\text{mod } 2$  加算手段、とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である。

【請求項 3】 符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号方法であり、 $K$ 、 $n$ 、 $m$  は自然数、 $K>1$ 、 $m>n$  であり、

$m$  ビットずつ受信した上記畳み込み符号の最も新しい連続する  $N$  ビット、 $N$  は  $N \geq m(K-1)/(m-n)$  を満足する予め決めた整数、を保持するステップと、

上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  ビットの保持されたデータからそれぞれ選択したデータを  $n$  個のグループで取り出すステップと、

上記  $n$  個の各グループ内の上記選択されたデータを  $\text{mod } 2$  加算して  $n$  ビットの復号データを出力するステップ、

とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N=mx$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である。

【請求項 4】 符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号回路の形成方法であり、 $K$ 、 $n$ 、 $m$  は自然数、 $K>1$ 、 $m>n$  であり、

上記原データが  $n$  ビット入力される毎に入力された最新の連続する  $K+n-1$  ビットの上記原データに対し上記  $m$  個の符号生成ベクトルを使って  $m$  ビットの符号データを生成することを繰り返し、各組が  $m$  ビットの符号デー

10

20

30

40

50

タを  $x$  組を得るステップと、 $x$  は  $x \geq (K-1)/(m-n)$  を満足する予め決めた整数であり、  
 各上記組の  $m$  ビットの符号データを生成する  $m$  個の符号生成多項式を上記  $m$  個の符号生成ベクトルの各  $(K+n-1)$  個の上記結合係数を含む  $N = x \cdot m$  個の結合係数により表現することにより、それぞれが  $N$  個の結合係数を有する  $N$  個の符号生成多項式を得るステップと、  
 上記  $N$  個の符号生成多項式の各  $N$  個の上記結合係数を要素とする  $N \times N$  の符号生成行列を形成するステップと、  
 上記符号生成行列からその逆行列を得るステップと、  
 上記逆行列の連続する  $n$  行を選択し、上記選択した各行の  $N$  個の要素を結合係数とし、上記  $N$  個の結合係数に従って、上記畳み込み符号データが入力される  $N$  個のシフト段と  $n$  個の  $\text{mod } 2$  加算器の対応する 1 つとを選択結合するステップ、とを含む。

【請求項 5】 請求項 1 に記載の簡易復号回路において、上記  $x$  の値は次式

$$x = \lceil (K-1)/(m-n) \rceil$$

によって与えられ、記号  $[p]$  は実数  $p$  以上の最小の整数を表す。

【請求項 6】 請求項 2 に記載の簡易復号回路において、上記  $N$  の値は次式

$$N = m \lceil (K-1)/(m-n) \rceil$$

によって与えられ、記号  $[p]$  は実数  $p$  以上の最小の整数を表す。

【請求項 7】 請求項 3 に記載の簡易復号方法において、上記  $N$  の値は次式

$$N = m \lceil (K-1)/(m-n) \rceil$$

によって与えられ、記号  $[p]$  は実数  $p$  以上の最小の整数を表す。

【請求項 8】 請求項 4 に記載の簡易復号回路の形成方法において、上記  $x$  の値は次式

$$x = \lceil (K-1)/(m-n) \rceil$$

によって与えられ、記号  $[p]$  は実数  $p$  以上の最小の整数を表す。

【請求項 9】 請求項 1 または 2 に記載の簡易復号回路において、 $K=4$ 、 $R=n/m=1/2$  であり、かつ上記復号生成行列の上記選択される行のベクトルが (1 1) である構成と、 $K=7$ 、 $R=n/m=1/2$  であり、かつ上記復号生成行列の上記選択される行のベクトルが (1110111010) である構成を除く。

【請求項 10】 請求項 3 に記載の簡易復号方法において、 $K=4$ 、 $R=n/m=1/2$  であり、かつ上記復号生成行列の上記選択される行のベクトルが (1 1) である場合と、 $K=7$ 、 $R=n/m=1/2$  であり、かつ上記復号生成行列の上記選択される行のベクトルが (1110111010) である場合を除く。

【請求項 11】 受信畳み込み符号を復号して推定原データを得るクレーム 1 または 2 の簡易復号回路と、  
 上記簡易復号回路からの推定原データを畳み込みにより

再符号化する、送信側符号器と同じ構成の畳み込み符号器と、

上記受信符号データと上記再符号化データとを排他的論理和により比較する比較手段と、

上記比較手段の比較結果から推定誤りを復号するビタビ復号回路と、

上記簡易復号回路からの推定原データと上記ビタビ復号回路からの推定誤りとを  $\text{mod } 2$  加算することにより上記推定原データの推定誤りを訂正して出力する  $\text{mod } 2$  加算手段、

とを含む復号器。

【請求項 12】 受信畳み込み符号を復号して推定原データを得るクレーム 1 または 2 の簡易復号回路と、

上記簡易復号回路からの推定原データを畳み込みにより再符号化する、送信側符号器と同じ構成の畳み込み符号器と、

上記受信符号データと上記再符号化データとを排他的論理和により比較して不一致を検出する比較手段と、

上記比較手段により検出された不一致の数を計数して、

その計数値に対応する評価値を出力する計数手段、

とを含む伝送路品質監視装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、誤り訂正符号の 1 つである畳み込み符号を受信して簡易に原データを復号する、それ自体では誤り訂正機能を持たない簡易復号方法及び簡易復号回路に関する。

【0002】

【従来の技術】 現在知られている畳み込み符号の復号方法の 1 つとして A. J. Viterbi により提案された復号アルゴリズムが最も高い誤り訂正能力と信頼性を有する復号方法であることが知られており、ビタビ復号器としてすでに衛星通信の分野で一部実用化され、その優位性が実証されている。しかし、ビタビ復号器は上述したような優れた特性を持つ反面、誤り訂正率を高めるためには回路規模が極めて大きくなり、しかもその結果、消費電力が大きくなるという欠点がある。

【0003】 一方、畳み込み符号を受信して誤り訂正復号を行う他の復号方法としては、閾値復号、逐次復号などがある。これらの復号方法もビタビ復号ほどではないにしてもそれを実施する復号回路の規模は大きく、その電力消費が大きくなる欠点がある。そこで、ビタビ復号器の低消費電力化及び回路規模の低減が可能な方式として S S T (Scarce State Transition) 方式が提案され、実用化されている (Ishitani et al, "A Scarce-State-Transition Viterbi-Decoder VLSI for Error Correction", IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. SC-22, NO. 4, AUGUST 1987, pp575-581)。

【0004】 図 1 は例えば上述の文献に示されているような、ブランチメトリック回路と A C S 回路とパスメモ

10

20

30

40

50

り回路からなる従来の狭義のビタビ復号回路16にSST方式を適用して構成したSST型ビタビ復号器の例であり、拘束長 $K=7$ 、符号化率 $1/2$ の場合である。SST型ビタビ復号器は、まず端子 $1_1$ 、 $1_1$ に与えられた伝送路誤りを含んでいるかも知れない受信畳み込み符号 $X_1$ 、 $X_2$ から誤り訂正機能を持たない簡易復号回路2により復号を行って原データDの推測復号データ $D'$ を得る。この文献に示された簡易復号回路は10個のシフト段に保持された10ビットの符号データから結合ベクトル(1110111010)で指定された7つのシフト段の出力符号データを選択してmod 2加算することにより原データを1ビット復号するものである。次に、この推測復号データ $D'$ を送信側畳み込み符号器と同じ構成の符号器3で再符号化し、この再符号化畳み込み符号 $X_1'$ 、 $X_2'$ と受信畳み込み符号データ(軟判定データ) $X_1$ 、 $X_2$ を比較器(排他的論理和回路)13<sub>1</sub>、13<sub>2</sub>で比較する。

【0005】伝送路誤りがなければ受信符号データ $X_1$ 、 $X_2$ と再符号化データ $X_1'$ 、 $X_2'$ は一致し、比較器13<sub>1</sub>、13<sub>2</sub>の出力は共に“0”となる。畳み込み符号データ $X_1$ 、 $X_2$ は原データの畳み込みの結果本来各データ $X_1$ 、 $X_2$ 、 $X_3$ 、…の前後複数ビットに渡って互いに影響を及ぼし合っているため、伝送路誤りが生じると正しい畳み込み符号では起こり得ないビットパターンが生じる。一方、畳み込み符号器3の出力再符号化データは、その入力データ $D'$ が推定誤りであったとしても畳み込み符号としての再符号化データには異常なビットパターンは生じない。従って、伝送路誤りがあった場合、比較器13<sub>1</sub>、13<sub>2</sub>には複数ビットに渡って連続または不連続に“1”が生じる。この様な推定誤りを表す比較結果を従来のビタビ復号回路14の入力として与えてその推定誤りを復号し、その出力と簡易復号回路2の復号結果 $D'$ とのmod 2加算をmod 2加算器15で行うことにより簡易復号結果 $D'$ の推定誤りを訂正し、原データDを得る。なお、遅延回路12<sub>1</sub>、12<sub>2</sub>、16はデータのタイミングを合わせるためのものである。

【0006】この様なSST型ビタビ復号器では、伝送路上で畳み込み符号データに誤りが生じなければ、狭義のビタビ復号回路14の入力は常に“0”である。そのため、ビタビ復号回路14は信号誤りのみを処理することになる。即ち、ビタビ復号回路14内のバスメモリ回路に記憶されるデータは、信号に誤りが生じたとき以外は“0”であり、バスメモリ回路のゲートのON-OFF動作は、ほとんど起こらない。一般に、CMOS回路は信号のON-OFFで電力を消費するため、SST型ビタビ復号器の消費電力は、従来型のビタビ復号器と比較すると大幅に低減される(復号後の誤り率 $P_e=10$ のとき約40%)。更に、従来型のビタビ復号器においては、所要バスメモリ長を削減するために最尤判定回路

を用いていたが、SST法を適用したビタビ復号器ではバスメモリ長が削減されるので最尤判定回路を省略することができる。これらの理由からSST型ビタビ復号器は、狭義のビタビ復号回路14の消費電力と回路規模の削減が可能であり、有効である。

【0007】送信側において従来用いられている符号化率 $R=1/2$ 、拘束長 $K=4$ の畳み込み符号器の最も簡単な構成例を図2に示す。この例では符号器は3段のシフトレジスタ4と、mod 2加算器5<sub>1</sub>、5<sub>2</sub>とから構成されている。原データD( $D_1$ 、 $D_2$ 、…)が入力端子6を通してシフトレジスタ4に順次入力され、mod 2加算器5<sub>1</sub>はシフトレジスタ4の全シフト段からの出力のmod 2加算を行い、mod 2加算器5<sub>2</sub>はシフトレジスタ4の第1及び第3シフト段からの出力をmod 2加算し、これらmod 2加算器5<sub>1</sub>、5<sub>2</sub>から畳み込み符号データ $X_1$ 、 $X_2$ が端子7<sub>1</sub>、7<sub>2</sub>に出力される。図1の受信側のSST型ビタビ復号器はこの符号データ $X_1$ 、 $X_2$ を受信し、復号する。図2の符号器においてシフトレジスタ4の第1段と第3段の出力は何れも常に両方のmod 2加算器5<sub>1</sub>、5<sub>2</sub>に与えられているので、データ $X_2$ に対するデータ $X_1$ の相対的關係はデータ $X_2$ を“0”に設定し、第2シフト段の出力をデータ $X_1$ としたときの関係と等価になる。従って、図2の符号器に対する図1内の簡易復号回路2は図3に示すように1つのmod 2加算器で構成できることは容易に理解される。この場合、入力符号データをmod 2加算器に結合する結合ベクトルは(11)で表される。

【0008】

【発明が解決しようとする課題】一般に符号の伝送能率を高めるためには符号化率 $R=n/m$ ( $n$ は入力情報ビット数、 $m$ は出力ビット数)を大きくし、即ち1に近づけ、また誤り訂正能力を高めるためには拘束長 $K$ をより長くすることが望まれる。しかしながら、この様なSST型ビタビ復号器で必要となる簡易復号回路2は、畳み込み符号の符号化率 $R$ が $1/2$ で拘束長 $K$ が7以下の場合には前述の文献に示されているように比較的容易に構成することができるが、一般に $n/m$ で表される任意の符号化率 $R$ の畳み込み符号を復号する簡易復号回路を構成するための論理設計アルゴリズムは見いだされていなかった。実際、符号化率 $R=1/2$ 以外の畳み込み符号に対する簡易復号回路はこの出願の発明者らにより符号化率 $R=3/4$ 、拘束長 $K=7$ の畳み込み符号について復号可能な簡易復号回路を試行錯誤的にたまたま構成することができただけである(川添等、「SST型高符号化率ビタビ復号器の検討」、1991年電子情報通信学会、秋期大会、B-156)。この様に拘束長 $K$ が7以下でかつ符号化率 $R$ が $1/2$ である符号以外の任意の符号化率の畳み込み符号に対する簡易復号回路を構成することは従来困難であり、従ってそのような符号に対する簡易復号回路を使用しなければならないSST型ビタビ

復号器は一般には構成することができなかった。そのため、符号化率  $n/m$  の符号に対しては S S T 型でない従来のビタビ復号器を用いる必要があり、上記の電力消費量が大きい問題点を解決できなかった。

【0009】この発明の第1の目的は、任意の拘束長と符号化率の畳み込み符号に対する簡易の復号方法を提供することである。この発明の第2の目的は、上記復号方法による簡易復号回路を提供することである。この発明の第3の目的は、上記簡易復号回路を形成する方法を提供することである。

【0010】この発明の第4の目的は、上記簡易復号回路を適用し、任意の符号化率と拘束長の畳み込み符号に対する消費電力の少ない S S T 型ビタビ復号器を提供することである。この発明の第5の目的は、上記簡易復号回路を適用した簡単な構成の伝送路品質監視装置を提供することである。

【0011】

【課題を解決するための手段】この発明によれば、第1の目的は符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号回路であり、 $K, n, m$  は自然数、 $K>1, m>n$  であり、順次受信した  $m$  ビット毎の上記畳み込み符号のデータがそれぞれ1ビットずつ供給され、それぞれが  $x \geq (K-1)/(m-n)$  を満足する予め決めた整数  $x$  個の直列接続されたシフト段を有する  $m$  個のシフトレジスタと、上記  $m$  個のシフトレジスタの全  $N=mx$  個のシフト段は  $N$  個の符号データを保持するためのシフトレジスタ回路を構成し、上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  個のシフト段から選択したそれぞれ複数の上記シフト段から出力を取り出す  $n$  個の選択結合手段と、上記  $n$  個の選択結合手段とそれぞれ接続され、それぞれ上記複数のシフト段の出力の  $\text{mod } 2$  加算を行い  $n$  ビットの復号データを出力する  $n$  個の  $\text{mod } 2$  加算手段、とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続する  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N=mx$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である、ように構成することにより達成することが出来る。

【0012】または符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルの

それぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号回路であり、 $K, n, m$  は自然数、 $K>1, m>n$  であり、受信した上記畳み込み符号のデータが1ビットずつ供給され、 $N \geq m(K-1)/(m-n)$  を満足する予め決めた整数  $N$  個の直列接続されたシフト段を有するシフトレジスタと、上記  $N$  個のシフト段は  $N$  個の符号データを保持するためのシフトレジスタ回路を構成し、上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  個のシフト段から選択したそれぞれ複数の上記シフト段から出力を取り出す  $n$  個の選択結合手段と、上記  $n$  個の選択結合手段とそれぞれ接続され、それぞれ上記複数のシフト段の出力の  $\text{mod } 2$  加算を行い  $n$  ビットの復号データを出力する  $n$  個の  $\text{mod } 2$  加算手段、とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である、ように構成することによって達成することが出来る。

【0013】この発明の第2の目的は、符号器に入力された原データの連続する  $K+n-1$  ビットから、それぞれが  $K+n-1$  個の予め決めた結合係数からなる  $m$  個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ  $\text{mod } 2$  加算して  $m$  ビットの符号データを出力することを、原データが  $n$  ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長  $K$ 、符号化率  $R=n/m$  の畳み込み符号を復号する簡易復号方法であり、 $K, n, m$  は自然数、 $K>1, m>n$  であり、 $m$  ビットずつ受信した上記畳み込み符号の最も新しい連続する  $N$  ビット、 $N$  は  $N \geq m(K-1)/(m-n)$  を満足する予め決めた整数、を保持するステップと、上記符号器の符号生成ベクトルに対応した復号生成行列中から選択された連続する  $n$  行がそれぞれ有する  $N$  個の要素である結合係数に従って上記  $N$  ビットの保持されたデータからそれぞれ選択したデータを  $n$  個のグループで取り出すステップと、上記  $n$  個の各グループ内の上記選択されたデータを  $\text{mod } 2$  加算して  $n$  ビットの復号データを出力するステップ、とを含み、上記復号生成行列は、 $n$  ビット毎に上記符号器に入力された原データの連続  $K+n-1$  ビットに対し上記符号器の  $m$  個の符号生成ベクトルを使って作られた  $N=mx$  個の符号を生成する  $N$  個の符号生成多項式の  $N \times N$  個の係数を要素とする正方行列の逆行列である、復号方法により達成することが出来る。

【0014】この発明の第3の目的は、符号器に入力された原データの連続する $K+n-1$ ビットから、それぞれが $K+n-1$ 個の予め決めた結合係数からなる $m$ 個の符号生成ベクトルのそれぞれの結合係数に従ってそれぞれ選択されたデータをそれぞれ $\text{mod } 2$ 加算して $m$ ビットの符号データを出力することを、原データが $n$ ビットずつシフトして上記符号器に入力される毎に繰り返して生成された拘束長 $K$ 、符号化率 $R=n/m$ の畳み込み符号を復号する簡易復号回路の形成方法であり、 $K$ 、 $n$ 、 $m$ は自然数、 $K>1$ 、 $m>n$ であり、上記原データが $n$ ビット入力される毎に入力された最新の連続する $K+n-1$ ビットの上記原データに対し上記 $m$ 個の符号生成ベクトルを使って $m$ ビットの符号データを生成することを繰り返し、各組が $m$ ビットの符号データを $x$ 組を得るステップと、 $x$ は $x \geq (K-1)/(m-n)$ を満足する予め決めた整数であり、各上記組の $m$ ビットの符号データを生成する $m$ 個の符号生成多項式を上記 $m$ 個の符号生成ベクトルの各 $(K+n-1)$ 個の上記結合係数を含む $N=xm$ 個の結合係数により表現することにより、それぞれが $N$ 個の結合係数を有する $N$ 個の符号生成多項式を得るステップと、上記 $N$ 個の符号生成多項式の各 $N$ 個の上記結合係数を要素とする $N \times N$ の符号生成行列を形成するステップと、上記符号生成行列からその逆行列を得るステップと、上記逆行列の連続する $n$ 行を選択し、上記選択した各行の $N$ 個の要素を結合係数とし、上記 $N$ 個の結合係数に従って、上記畳み込み符号データが入力される $N$ 個のシフト段と $n$ 個の $\text{mod } 2$ 加算器の対応する1つとを選択結合するステップ、とを含む、方法により達成することが出来る。

【0015】この発明の第4の目的は、上記第1または第2の目的を達成し、受信畳み込み符号を復号して推定原データを得る上記簡易復号回路と、上記簡易復号回路からの推定原データを畳み込みにより再符号化する、送信側符号器と同じ構成の畳み込み符号器と、上記受信符号データと上記再符号化データとを排他的論理和により比較する比較手段と、上記比較手段の比較結果から推定誤りを復号するビット復号回路と、上記簡易復号回路からの推定原データと上記ビット復号回路からの推定誤りとを $\text{mod } 2$ 加算することにより上記推定原データの推定誤りを訂正して出力する $\text{mod } 2$ 加算手段、とを含む復号器によって達成することが出来る。

【0016】この発明の第5の目的は、上記第1または第2の目的を達成し、受信畳み込み符号を復号して推定原データを得る上記簡易復号回路と、上記簡易復号回路からの推定原データを畳み込みにより再符号化する、送信側符号器と同じ構成の畳み込み符号器と、上記受信符号データと上記再符号化データとを排他的論理和により比較して不一致を検出する比較手段と、上記比較手段により検出された不一致の数を計数して、その計数値に対応する評価値を出力する計数手段、とを含む伝送路品質

監視装置によって達成することが出来る。

【0017】

【実施例】まずこの発明に到達する過程を理解し易くするため、この発明の簡易復号回路が対象とする符号化率 $n/m$ の畳み込み符号の生成について説明する。拘束長 $K$ で所望の符号化率 $R=n/m$ の畳み込み符号を生成する符号器は周知である。例えばバンクチャード型符号器の場合は、図4に $K=7$ 、 $R=n/m=3/4$ の場合を示すように、7つのシフト段を有するシフトレジスタ4と、これら7つのシフト段の予め決めたものと結合された $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ 、及び $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ の出力から $n=3$ ビット毎に予め決めた $m/2=2$ 個のビット位置のデータを抽出するシフトレジスタ $8_1$ 、 $8_2$ から成るバンクチャード回路8とから構成される。7つのシフト段と各 $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ との結合状態は7つの係数0または1を要素とする結合ベクトルで表され、図4の場合の結合ベクトルは(1101101)と(1001101)であり、係数“1”及び“0”はそれぞれ結合、非結合を表している。なお、拘束長 $K$ は1つの $\text{mod } 2$ 加算器と係数“1”で結合された複数のシフト段の一端から他端までの段数に対応している。

【0018】 $K$ シフト段と各 $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ との結合は結合ベクトルの係数(要素)が“1”に対応するシフト段と $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ との結線として示してある。この結合ベクトルとバンクチャード回路8の抽出ビット位置は、生成される畳み込み符号の誤り訂正能力が最大となるように選ばれた場合を示している。この例では原データ $D$ が3ビット入力される毎に4ビットの符号化データ $I_1$ 、 $I_2$ 、 $Q_1$ 、 $Q_2$ が出力される。即ち、シフトレジスタ4の全シフト段に原データ $D_1 \sim D_7$ が保持された状態で $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ からデータ対 $I_1$ 、 $Q_1$ が出力され、次にデータ $D_8$ が入力されると $\text{mod } 2$ 加算器 $5_1$ 、 $5_2$ からデータ対 $I_2$ 、 $Q_2$ が出力され、以下同様にデータ対 $(I_1, Q_1)$ 、 $(I_2, Q_2)$ 、 $(I_3, Q_3)$ 、 $(I_4, Q_4)$ 、 $(I_5, Q_5)$ 、 $(I_6, Q_6)$ 、 $(I_7, Q_7)$ 、 $(I_8, Q_8)$ 、 $(I_9, Q_9)$ 、 $(I_{10}, Q_{10})$ 、 $(I_{11}, Q_{11})$ 、 $(I_{12}, Q_{12})$ 、 $(I_{13}, Q_{13})$ 、 $(I_{14}, Q_{14})$ 、 $(I_{15}, Q_{15})$ 、 $(I_{16}, Q_{16})$ 、 $(I_{17}, Q_{17})$ 、 $(I_{18}, Q_{18})$ 、 $(I_{19}, Q_{19})$ 、 $(I_{20}, Q_{20})$ 、 $(I_{21}, Q_{21})$ 、 $(I_{22}, Q_{22})$ 、 $(I_{23}, Q_{23})$ 、 $(I_{24}, Q_{24})$ 、 $(I_{25}, Q_{25})$ 、 $(I_{26}, Q_{26})$ 、 $(I_{27}, Q_{27})$ 、 $(I_{28}, Q_{28})$ 、 $(I_{29}, Q_{29})$ 、 $(I_{30}, Q_{30})$ 、 $(I_{31}, Q_{31})$ 、 $(I_{32}, Q_{32})$ 、 $(I_{33}, Q_{33})$ 、 $(I_{34}, Q_{34})$ 、 $(I_{35}, Q_{35})$ 、 $(I_{36}, Q_{36})$ 、 $(I_{37}, Q_{37})$ 、 $(I_{38}, Q_{38})$ 、 $(I_{39}, Q_{39})$ 、 $(I_{40}, Q_{40})$ 、 $(I_{41}, Q_{41})$ 、 $(I_{42}, Q_{42})$ 、 $(I_{43}, Q_{43})$ 、 $(I_{44}, Q_{44})$ 、 $(I_{45}, Q_{45})$ 、 $(I_{46}, Q_{46})$ 、 $(I_{47}, Q_{47})$ 、 $(I_{48}, Q_{48})$ 、 $(I_{49}, Q_{49})$ 、 $(I_{50}, Q_{50})$ 、 $(I_{51}, Q_{51})$ 、 $(I_{52}, Q_{52})$ 、 $(I_{53}, Q_{53})$ 、 $(I_{54}, Q_{54})$ 、 $(I_{55}, Q_{55})$ 、 $(I_{56}, Q_{56})$ 、 $(I_{57}, Q_{57})$ 、 $(I_{58}, Q_{58})$ 、 $(I_{59}, Q_{59})$ 、 $(I_{60}, Q_{60})$ 、 $(I_{61}, Q_{61})$ 、 $(I_{62}, Q_{62})$ 、 $(I_{63}, Q_{63})$ 、 $(I_{64}, Q_{64})$ 、 $(I_{65}, Q_{65})$ 、 $(I_{66}, Q_{66})$ 、 $(I_{67}, Q_{67})$ 、 $(I_{68}, Q_{68})$ 、 $(I_{69}, Q_{69})$ 、 $(I_{70}, Q_{70})$ 、 $(I_{71}, Q_{71})$ 、 $(I_{72}, Q_{72})$ 、 $(I_{73}, Q_{73})$ 、 $(I_{74}, Q_{74})$ 、 $(I_{75}, Q_{75})$ 、 $(I_{76}, Q_{76})$ 、 $(I_{77}, Q_{77})$ 、 $(I_{78}, Q_{78})$ 、 $(I_{79}, Q_{79})$ 、 $(I_{80}, Q_{80})$ 、 $(I_{81}, Q_{81})$ 、 $(I_{82}, Q_{82})$ 、 $(I_{83}, Q_{83})$ 、 $(I_{84}, Q_{84})$ 、 $(I_{85}, Q_{85})$ 、 $(I_{86}, Q_{86})$ 、 $(I_{87}, Q_{87})$ 、 $(I_{88}, Q_{88})$ 、 $(I_{89}, Q_{89})$ 、 $(I_{90}, Q_{90})$ 、 $(I_{91}, Q_{91})$ 、 $(I_{92}, Q_{92})$ 、 $(I_{93}, Q_{93})$ 、 $(I_{94}, Q_{94})$ 、 $(I_{95}, Q_{95})$ 、 $(I_{96}, Q_{96})$ 、 $(I_{97}, Q_{97})$ 、 $(I_{98}, Q_{98})$ 、 $(I_{99}, Q_{99})$ 、 $(I_{100}, Q_{100})$ 、 $(I_{101}, Q_{101})$ 、 $(I_{102}, Q_{102})$ 、 $(I_{103}, Q_{103})$ 、 $(I_{104}, Q_{104})$ 、 $(I_{105}, Q_{105})$ 、 $(I_{106}, Q_{106})$ 、 $(I_{107}, Q_{107})$ 、 $(I_{108}, Q_{108})$ 、 $(I_{109}, Q_{109})$ 、 $(I_{110}, Q_{110})$ 、 $(I_{111}, Q_{111})$ 、 $(I_{112}, Q_{112})$ 、 $(I_{113}, Q_{113})$ 、 $(I_{114}, Q_{114})$ 、 $(I_{115}, Q_{115})$ 、 $(I_{116}, Q_{116})$ 、 $(I_{117}, Q_{117})$ 、 $(I_{118}, Q_{118})$ 、 $(I_{119}, Q_{119})$ 、 $(I_{120}, Q_{120})$ 、 $(I_{121}, Q_{121})$ 、 $(I_{122}, Q_{122})$ 、 $(I_{123}, Q_{123})$ 、 $(I_{124}, Q_{124})$ 、 $(I_{125}, Q_{125})$ 、 $(I_{126}, Q_{126})$ 、 $(I_{127}, Q_{127})$ 、 $(I_{128}, Q_{128})$ 、 $(I_{129}, Q_{129})$ 、 $(I_{130}, Q_{130})$ 、 $(I_{131}, Q_{131})$ 、 $(I_{132}, Q_{132})$ 、 $(I_{133}, Q_{133})$ 、 $(I_{134}, Q_{134})$ 、 $(I_{135}, Q_{135})$ 、 $(I_{136}, Q_{136})$ 、 $(I_{137}, Q_{137})$ 、 $(I_{138}, Q_{138})$ 、 $(I_{139}, Q_{139})$ 、 $(I_{140}, Q_{140})$ 、 $(I_{141}, Q_{141})$ 、 $(I_{142}, Q_{142})$ 、 $(I_{143}, Q_{143})$ 、 $(I_{144}, Q_{144})$ 、 $(I_{145}, Q_{145})$ 、 $(I_{146}, Q_{146})$ 、 $(I_{147}, Q_{147})$ 、 $(I_{148}, Q_{148})$ 、 $(I_{149}, Q_{149})$ 、 $(I_{150}, Q_{150})$ 、 $(I_{151}, Q_{151})$ 、 $(I_{152}, Q_{152})$ 、 $(I_{153}, Q_{153})$ 、 $(I_{154}, Q_{154})$ 、 $(I_{155}, Q_{155})$ 、 $(I_{156}, Q_{156})$ 、 $(I_{157}, Q_{157})$ 、 $(I_{158}, Q_{158})$ 、 $(I_{159}, Q_{159})$ 、 $(I_{160}, Q_{160})$ 、 $(I_{161}, Q_{161})$ 、 $(I_{162}, Q_{162})$ 、 $(I_{163}, Q_{163})$ 、 $(I_{164}, Q_{164})$ 、 $(I_{165}, Q_{165})$ 、 $(I_{166}, Q_{166})$ 、 $(I_{167}, Q_{167})$ 、 $(I_{168}, Q_{168})$ 、 $(I_{169}, Q_{169})$ 、 $(I_{170}, Q_{170})$ 、 $(I_{171}, Q_{171})$ 、 $(I_{172}, Q_{172})$ 、 $(I_{173}, Q_{173})$ 、 $(I_{174}, Q_{174})$ 、 $(I_{175}, Q_{175})$ 、 $(I_{176}, Q_{176})$ 、 $(I_{177}, Q_{177})$ 、 $(I_{178}, Q_{178})$ 、 $(I_{179}, Q_{179})$ 、 $(I_{180}, Q_{180})$ 、 $(I_{181}, Q_{181})$ 、 $(I_{182}, Q_{182})$ 、 $(I_{183}, Q_{183})$ 、 $(I_{184}, Q_{184})$ 、 $(I_{185}, Q_{185})$ 、 $(I_{186}, Q_{186})$ 、 $(I_{187}, Q_{187})$ 、 $(I_{188}, Q_{188})$ 、 $(I_{189}, Q_{189})$ 、 $(I_{190}, Q_{190})$ 、 $(I_{191}, Q_{191})$ 、 $(I_{192}, Q_{192})$ 、 $(I_{193}, Q_{193})$ 、 $(I_{194}, Q_{194})$ 、 $(I_{195}, Q_{195})$ 、 $(I_{196}, Q_{196})$ 、 $(I_{197}, Q_{197})$ 、 $(I_{198}, Q_{198})$ 、 $(I_{199}, Q_{199})$ 、 $(I_{200}, Q_{200})$ 、 $(I_{201}, Q_{201})$ 、 $(I_{202}, Q_{202})$ 、 $(I_{203}, Q_{203})$ 、 $(I_{204}, Q_{204})$ 、 $(I_{205}, Q_{205})$ 、 $(I_{206}, Q_{206})$ 、 $(I_{207}, Q_{207})$ 、 $(I_{208}, Q_{208})$ 、 $(I_{209}, Q_{209})$ 、 $(I_{210}, Q_{210})$ 、 $(I_{211}, Q_{211})$ 、 $(I_{212}, Q_{212})$ 、 $(I_{213}, Q_{213})$ 、 $(I_{214}, Q_{214})$ 、 $(I_{215}, Q_{215})$ 、 $(I_{216}, Q_{216})$ 、 $(I_{217}, Q_{217})$ 、 $(I_{218}, Q_{218})$ 、 $(I_{219}, Q_{219})$ 、 $(I_{220}, Q_{220})$ 、 $(I_{221}, Q_{221})$ 、 $(I_{222}, Q_{222})$ 、 $(I_{223}, Q_{223})$ 、 $(I_{224}, Q_{224})$ 、 $(I_{225}, Q_{225})$ 、 $(I_{226}, Q_{226})$ 、 $(I_{227}, Q_{227})$ 、 $(I_{228}, Q_{228})$ 、 $(I_{229}, Q_{229})$ 、 $(I_{230}, Q_{230})$ 、 $(I_{231}, Q_{231})$ 、 $(I_{232}, Q_{232})$ 、 $(I_{233}, Q_{233})$ 、 $(I_{234}, Q_{234})$ 、 $(I_{235}, Q_{235})$ 、 $(I_{236}, Q_{236})$ 、 $(I_{237}, Q_{237})$ 、 $(I_{238}, Q_{238})$ 、 $(I_{239}, Q_{239})$ 、 $(I_{240}, Q_{240})$ 、 $(I_{241}, Q_{241})$ 、 $(I_{242}, Q_{242})$ 、 $(I_{243}, Q_{243})$ 、 $(I_{244}, Q_{244})$ 、 $(I_{245}, Q_{245})$ 、 $(I_{246}, Q_{246})$ 、 $(I_{247}, Q_{247})$ 、 $(I_{248}, Q_{248})$ 、 $(I_{249}, Q_{249})$ 、 $(I_{250}, Q_{250})$ 、 $(I_{251}, Q_{251})$ 、 $(I_{252}, Q_{252})$ 、 $(I_{253}, Q_{253})$ 、 $(I_{254}, Q_{254})$ 、 $(I_{255}, Q_{255})$ 、 $(I_{256}, Q_{256})$ 、 $(I_{257}, Q_{257})$ 、 $(I_{258}, Q_{258})$ 、 $(I_{259}, Q_{259})$ 、 $(I_{260}, Q_{260})$ 、 $(I_{261}, Q_{261})$ 、 $(I_{262}, Q_{262})$ 、 $(I_{263}, Q_{263})$ 、 $(I_{264}, Q_{264})$ 、 $(I_{265}, Q_{265})$ 、 $(I_{266}, Q_{266})$ 、 $(I_{267}, Q_{267})$ 、 $(I_{268}, Q_{268})$ 、 $(I_{269}, Q_{269})$ 、 $(I_{270}, Q_{270})$ 、 $(I_{271}, Q_{271})$ 、 $(I_{272}, Q_{272})$ 、 $(I_{273}, Q_{273})$ 、 $(I_{274}, Q_{274})$ 、 $(I_{275}, Q_{275})$ 、 $(I_{276}, Q_{276})$ 、 $(I_{277}, Q_{277})$ 、 $(I_{278}, Q_{278})$ 、 $(I_{279}, Q_{279})$ 、 $(I_{280}, Q_{280})$ 、 $(I_{281}, Q_{281})$ 、 $(I_{282}, Q_{282})$ 、 $(I_{283}, Q_{283})$ 、 $(I_{284}, Q_{284})$ 、 $(I_{285}, Q_{285})$ 、 $(I_{286}, Q_{286})$ 、 $(I_{287}, Q_{287})$ 、 $(I_{288}, Q_{288})$ 、 $(I_{289}, Q_{289})$ 、 $(I_{290}, Q_{290})$ 、 $(I_{291}, Q_{291})$ 、 $(I_{292}, Q_{292})$ 、 $(I_{293}, Q_{293})$ 、 $(I_{294}, Q_{294})$ 、 $(I_{295}, Q_{295})$ 、 $(I_{296}, Q_{296})$ 、 $(I_{297}, Q_{297})$ 、 $(I_{298}, Q_{298})$ 、 $(I_{299}, Q_{299})$ 、 $(I_{300}, Q_{300})$ 、 $(I_{301}, Q_{301})$ 、 $(I_{302}, Q_{302})$ 、 $(I_{303}, Q_{303})$ 、 $(I_{304}, Q_{304})$ 、 $(I_{305}, Q_{305})$ 、 $(I_{306}, Q_{306})$ 、 $(I_{307}, Q_{307})$ 、 $(I_{308}, Q_{308})$ 、 $(I_{309}, Q_{309})$ 、 $(I_{310}, Q_{310})$ 、 $(I_{311}, Q_{311})$ 、 $(I_{312}, Q_{312})$ 、 $(I_{313}, Q_{313})$ 、 $(I_{314}, Q_{314})$ 、 $(I_{315}, Q_{315})$ 、 $(I_{316}, Q_{316})$ 、 $(I_{317}, Q_{317})$ 、 $(I_{318}, Q_{318})$ 、 $(I_{319}, Q_{319})$ 、 $(I_{320}, Q_{320})$ 、 $(I_{321}, Q_{321})$ 、 $(I_{322}, Q_{322})$ 、 $(I_{323}, Q_{323})$ 、 $(I_{324}, Q_{324})$ 、 $(I_{325}, Q_{325})$ 、 $(I_{326}, Q_{326})$ 、 $(I_{327}, Q_{327})$ 、 $(I_{328}, Q_{328})$ 、 $(I_{329}, Q_{329})$ 、 $(I_{330}, Q_{330})$ 、 $(I_{331}, Q_{331})$ 、 $(I_{332}, Q_{332})$ 、 $(I_{333}, Q_{333})$ 、 $(I_{334}, Q_{334})$ 、 $(I_{335}, Q_{335})$ 、 $(I_{336}, Q_{336})$ 、 $(I_{337}, Q_{337})$ 、 $(I_{338}, Q_{338})$ 、 $(I_{339}, Q_{339})$ 、 $(I_{340}, Q_{340})$ 、 $(I_{341}, Q_{341})$ 、 $(I_{342}, Q_{342})$ 、 $(I_{343}, Q_{343})$ 、 $(I_{344}, Q_{344})$ 、 $(I_{345}, Q_{345})$ 、 $(I_{346}, Q_{346})$ 、 $(I_{347}, Q_{347})$ 、 $(I_{348}, Q_{348})$ 、 $(I_{349}, Q_{349})$ 、 $(I_{350}, Q_{350})$ 、 $(I_{351}, Q_{351})$ 、 $(I_{352}, Q_{352})$ 、 $(I_{353}, Q_{353})$ 、 $(I_{354}, Q_{354})$ 、 $(I_{355}, Q_{355})$ 、 $(I_{356}, Q_{356})$ 、 $(I_{357}, Q_{357})$ 、 $(I_{358}, Q_{358})$ 、 $(I_{359}, Q_{359})$ 、 $(I_{360}, Q_{360})$ 、 $(I_{361}, Q_{361})$ 、 $(I_{362}, Q_{362})$ 、 $(I_{363}, Q_{363})$ 、 $(I_{364}, Q_{364})$ 、 $(I_{365}, Q_{365})$ 、 $(I_{366}, Q_{366})$ 、 $(I_{367}, Q_{367})$ 、 $(I_{368}, Q_{368})$ 、 $(I_{369}, Q_{369})$ 、 $(I_{370}, Q_{370})$ 、 $(I_{371}, Q_{371})$ 、 $(I_{372}, Q_{372})$ 、 $(I_{373}, Q_{373})$ 、 $(I_{374}, Q_{374})$ 、 $(I_{375}, Q_{375})$ 、 $(I_{376}, Q_{376})$ 、 $(I_{377}, Q_{377})$ 、 $(I_{378}, Q_{378})$ 、 $(I_{379}, Q_{379})$ 、 $(I_{380}, Q_{380})$ 、 $(I_{381}, Q_{381})$ 、 $(I_{382}, Q_{382})$ 、 $(I_{383}, Q_{383})$ 、 $(I_{384}, Q_{384})$ 、 $(I_{385}, Q_{385})$ 、 $(I_{386}, Q_{386})$ 、 $(I_{387}, Q_{387})$ 、 $(I_{388}, Q_{388})$ 、 $(I_{389}, Q_{389})$ 、 $(I_{390}, Q_{390})$ 、 $(I_{391}, Q_{391})$ 、 $(I_{392}, Q_{392})$ 、 $(I_{393}, Q_{393})$ 、 $(I_{394}, Q_{394})$ 、 $(I_{395}, Q_{395})$ 、 $(I_{396}, Q_{396})$ 、 $(I_{397}, Q_{397})$ 、 $(I_{398}, Q_{398})$ 、 $(I_{399}, Q_{399})$ 、 $(I_{400}, Q_{400})$ 、 $(I_{401}, Q_{401})$ 、 $(I_{402}, Q_{402})$ 、 $(I_{403}, Q_{403})$ 、 $(I_{404}, Q_{404})$ 、 $(I_{405}, Q_{405})$ 、 $(I_{406}, Q_{406})$ 、 $(I_{407}, Q_{407})$ 、 $(I_{408}, Q_{408})$ 、 $(I_{409}, Q_{409})$ 、 $(I_{410}, Q_{410})$ 、 $(I_{411}, Q_{411})$ 、 $(I_{412}, Q_{412})$ 、 $(I_{413}, Q_{413})$ 、 $(I_{414}, Q_{414})$ 、 $(I_{415}, Q_{415})$ 、 $(I_{416}, Q_{416})$ 、 $(I_{417}, Q_{417})$ 、 $(I_{418}, Q_{418})$ 、 $(I_{419}, Q_{419})$ 、 $(I_{420}, Q_{420})$ 、 $(I_{421}, Q_{421})$ 、 $(I_{422}, Q_{422})$ 、 $(I_{423}, Q_{423})$ 、 $(I_{424}, Q_{424})$ 、 $(I_{425}, Q_{425})$ 、 $(I_{426}, Q_{426})$ 、 $(I_{427}, Q_{427})$ 、 $(I_{428}, Q_{428})$ 、 $(I_{429}, Q_{429})$ 、 $(I_{430}, Q_{430})$ 、 $(I_{431}, Q_{431})$ 、 $(I_{432}, Q_{432})$ 、 $(I_{433}, Q_{433})$ 、 $(I_{434}, Q_{434})$ 、 $(I_{435}, Q_{435})$ 、 $(I_{436}, Q_{436})$ 、 $(I_{437}, Q_{437})$ 、 $(I_{438}, Q_{438})$ 、 $(I_{439}, Q_{439})$ 、 $(I_{440}, Q_{440})$ 、 $(I_{441}, Q_{441})$ 、 $(I_{442}, Q_{442})$ 、 $(I_{443}, Q_{443})$ 、 $(I_{444}, Q_{444})$ 、 $(I_{445}, Q_{445})$ 、 $(I_{446}, Q_{446})$ 、 $(I_{447}, Q_{447})$ 、 $(I_{448}, Q_{448})$ 、 $(I_{449}, Q_{449})$ 、 $(I_{450}, Q_{450})$ 、 $(I_{451}, Q_{451})$ 、 $(I_{452}, Q_{452})$ 、 $(I_{453}, Q_{453})$ 、 $(I_{454}, Q_{454})$ 、 $(I_{455}, Q_{455})$ 、 $(I_{456}, Q_{456})$ 、 $(I_{457}, Q_{457})$ 、 $(I_{458}, Q_{458})$ 、 $(I_{459}, Q_{459})$ 、 $(I_{460}, Q_{460})$ 、 $(I_{461}, Q_{461})$ 、 $(I_{462}, Q_{462})$ 、 $(I_{463}, Q_{463})$ 、 $(I_{464}, Q_{464})$ 、 $(I_{465}, Q_{465})$ 、 $(I_{466}, Q_{466})$ 、 $(I_{467}, Q_{467})$ 、 $(I_{468}, Q_{468})$ 、 $(I_{469}, Q_{469})$ 、 $(I_{470}, Q_{470})$ 、 $(I_{471}, Q_{471})$ 、 $(I_{472}, Q_{472})$ 、 $(I_{473}, Q_{473})$ 、 $(I_{474}, Q_{474})$ 、 $(I_{475}, Q_{475})$ 、 $(I_{476}, Q_{476})$ 、 $(I_{477}, Q_{477})$ 、 $(I_{478}, Q_{478})$ 、 $(I_{479}, Q_{479})$ 、 $(I_{480}, Q_{480})$ 、 $(I_{481}, Q_{481})$ 、 $(I_{482}, Q_{482})$ 、 $(I_{483}, Q_{483})$ 、 $(I_{484}, Q_{484})$ 、 $(I_{485}, Q_{485})$ 、 $(I_{486}, Q_{486})$ 、 $(I_{487}, Q_{487})$ 、 $(I_{488}, Q_{488})$ 、 $(I_{489}, Q_{489})$ 、 $(I_{490}, Q_{490})$ 、 $(I_{491}, Q_{491})$ 、 $(I_{492}, Q_{492})$ 、 $(I_{493}, Q_{493})$ 、 $(I_{494}, Q_{494})$ 、 $(I_{495}, Q_{495})$ 、 $(I_{496}, Q_{496})$ 、 $(I_{497}, Q_{497})$ 、 $(I_{498}, Q_{498})$ 、 $(I_{499}, Q_{499})$ 、 $(I_{500}, Q_{500})$ 、 $(I_{501}, Q_{501})$ 、 $(I_{502}, Q_{502})$ 、 $(I_{503}, Q_{503})$ 、 $(I_{504}, Q_{504})$ 、 $(I_{505}, Q_{505})$ 、 $(I_{506}, Q_{506})$ 、 $(I_{507}, Q_{507})$ 、 $(I_{508}, Q_{508})$ 、 $(I_{509}, Q_{509})$ 、 $(I_{510}, Q_{510})$ 、 $(I_{511}, Q_{511})$ 、 $(I_{512}, Q_{512})$ 、 $(I_{513}, Q_{513})$ 、 $(I_{514}, Q_{514})$ 、 $(I_{515}, Q_{515})$ 、 $(I_{516}, Q_{516})$ 、 $(I_{517}, Q_{517})$ 、 $(I_{518}, Q_{518})$ 、 $(I_{519}, Q_{519})$ 、 $(I_{520}, Q_{520})$ 、 $(I_{521}, Q_{521})$ 、 $(I_{522}, Q_{522})$ 、 $(I_{523}, Q_{523})$ 、 $(I_{524}, Q_{524})$ 、 $(I_{525}, Q_{525})$ 、 $(I_{526}, Q_{526})$ 、 $(I_{527}, Q_{527})$ 、 $(I_{528}, Q_{528})$ 、 $(I_{529}, Q_{529})$ 、 $(I_{530}, Q_{530})$ 、 $(I_{531}, Q_{531})$ 、 $(I_{532}, Q_{532})$ 、 $(I_{533}, Q_{533})$ 、 $(I_{534}, Q_{534})$ 、 $(I_{535}, Q_{535})$ 、 $(I_{536}, Q_{536})$ 、 $(I_{537}, Q_{537})$ 、 $(I_{538}, Q_{538})$ 、 $(I_{539}, Q_{539})$ 、 $(I_{540}, Q_{540})$ 、 $(I_{541}, Q_{541})$ 、 $(I_{542}, Q_{542})$ 、 $(I_{543}, Q_{543})$ 、 $(I_{544}, Q_{544})$ 、 $(I_{545}, Q_{545})$ 、 $(I_{546}, Q_{546})$ 、 $($

対しデータ  $I_1$ 、 $Q_1$  を生成する2つの符号生成ベクトル（各9ビット）と、データ  $I_1$  を生成する1つの符号生成ベクトル（9ビット）と、データ  $Q_1$  を生成する1つの符号生成ベクトル（9ビット）を規定することができる。

【0020】図5は図4と同様に拘束長  $K=7$ 、符号化率  $R=3/4$  の場合のバンクチュアド型でない畳み込み符号器の例であり、9個のシフト段を有するシフトレジスタ4と、9個のシフト段とそれぞれ予め決めた結合ベクトルで結合された4つの  $\text{mod } 2$  加算器  $5_1 \sim 5_4$  とから構成されている。図5の接続から明らかなように  $\text{mod } 2$  加算器  $5_1$  とシフトレジスタ4の接続は  $\text{mod } 2$  加算器  $5_1$  とシフトレジスタ4の接続と接続位置が1段ずれているだけであり、同様に  $\text{mod } 2$  加算器  $5_2$  とシフトレジスタ4の接続は  $\text{mod } 2$  加算器  $5_2$  と2段ずれているだけである。従って、図5の符号器は図4の符号器と本質的に同じものであり、何れの場合も4ビットの

$$\begin{aligned} I_1' &= D_1 + D_2 + D_3 + D_4 + D_5, \\ I_2' &= D_2 + D_3 + D_4 + D_5 + D_6, \\ Q_1' &= D_1 + D_3 + D_4 + D_5 + D_6, \\ Q_2' &= D_2 + D_4 + D_5 + D_6 + D_7, \end{aligned}$$

の様に表示することができる。但し上式中の記号“+”は  $\text{mod } 2$  加算（排他的論理和）を表すものとする。前述のように、受信側において受信した  $K=7$ 、 $R=3/4$  の畳み込み符号データ  $I_1'$ 、 $I_2'$ 、 $Q_1'$ 、 $Q_2'$  から原データ  $D_1 \sim D_7$  を復号できる誤り訂正機能を持たない簡易復号回路はこの出願の発明者らにより試行錯誤的にたまたま1つだけ構成することができたが、一般に拘束長

$$\begin{aligned} I_1^2 &= D_1 + D_2 + D_3 + D_4 + D_5, \\ I_2^2 &= D_2 + D_3 + D_4 + D_5 + D_6, \\ Q_1^2 &= D_1 + D_3 + D_4 + D_5 + D_6, \\ Q_2^2 &= D_2 + D_4 + D_5 + D_6 + D_7, \end{aligned}$$

で表される。式(1)では9個の変数  $D_1 \sim D_7$  により1組4個の  $\text{mod } 2$  加算値  $I_1^1$ 、 $I_2^1$ 、 $Q_1^1$ 、 $Q_2^1$  を決定する1組4つの多項式が得られ、式(2)では3個の変数  $D_6 \sim D_7$  が新たに追加されて更に4個の  $\text{mod } 2$  加算値を決定する4つの多項式が得られる。この様に原データ  $D$  が3ビットずつ入力される毎に1組4個ずつ多

$$4x \geq 9 + 3(x-1)$$

が成立するので、式(3)を満足する最小の整数  $x$  の値は  $x=6$  となり、24個の変数  $D_1 \sim D_{24}$  を含む24個の多項式が得られる。このことは畳み込み符号データである24個（6組）の  $\text{mod } 2$  加算値  $(I_1^1, I_2^1, Q_1^1, Q_2^1)$ 、 $(I_1^2, I_2^2, Q_1^2, Q_2^2)$ 、 $\dots$ 、 $(I_1^6, I_2^6, Q_1^6, Q_2^6)$  が与えられれば逆に変数  $D_1 \sim D_{24}$  を決定することができることを意味している。

そこで受信側の簡易復号回路においては少なくとも24

$$\begin{aligned} I_1^1 &= a_{1,1} D_1 + a_{1,2} D_2 + \dots + a_{1,24} D_{24}, \\ I_2^1 &= a_{2,1} D_1 + a_{2,2} D_2 + \dots + a_{2,24} D_{24}, \\ Q_1^1 &= a_{3,1} D_1 + a_{3,2} D_2 + \dots + a_{3,24} D_{24}, \end{aligned}$$

畳み込み符号  $I_1'$ 、 $I_2'$ 、 $Q_1'$ 、 $Q_2'$  を得るのに連続9ビットの入力データを必要とし、原データ  $D$  が3ビット追加される毎に1組4ビット  $(I_1'$ 、 $I_2'$ 、 $Q_1'$ 、 $Q_2'$ ) の畳み込み符号が出力される。従って、図5の場合これら4ビットの符号データを生成するための4つの符号生成ベクトル（各9ビット）は上記4つの結合ベクトルと同じものであり、これらは図4の4つの符号生成ベクトルと同じである。一般に、拘束長  $K$ 、符号化率  $n/m$  の場合、 $m$  ビット1組の畳み込み符号を得るには連続  $K+n-1$  ビットの入力データを必要とし、入力データ  $D$  が  $n$  ビット追加される毎に  $m$  ビットの畳み込み符号を得る  $K+n-1$  ビットの生成ベクトルを  $n$  個規定することができる。

【0021】図4及び5で示した例では9ビットの原データ  $D_1 \sim D_9$  を用いて最初の1組の符号データ  $I_1^1$ 、 $I_2^1$ 、 $Q_1^1$ 、 $Q_2^1$  をそれぞれ次式

(1)

$K$ 、符号化率  $R=n/m$  の畳み込み符号をビット復号アルゴリズムを使わないで復号できる簡易復号回路を構成することは非常に困難であった。

【0022】ところで、式(1)で表される状態にある図4又は5の符号器に更に3ビットの原データ  $D_8$ 、 $D_{11}$ 、 $D_{12}$  を入力して得られる符号器の1組の出力符号データ  $I_1^2$ 、 $I_2^2$ 、 $Q_1^2$ 、 $Q_2^2$  は次式

(2)

項式が増加するので、これを繰り返せばある時点で得られた多項式の数が使用された変数  $D_1$ 、 $D_2$ 、 $\dots$  の数と等しいかそれより大きくなる。例えば上述の  $K=7$ 、 $n/m=3/4$  の場合は、全部で  $x$  組の多項式が得られた時点で全多項式の数が変数データ  $D$  の数以上になるとすると、次式

(3)

40 個の受信符号データを保持することができるシフトレジスタを設ければ、このシフトレジスタに保持された24個の受信データ、即ち6組の受信畳み込み符号から24ビットの原データ  $D_1 \sim D_{24}$  を以下のようにして求めることができる。

【0023】上述の24個の多項式を、係数  $a_{i,j}$  ( $a_{i,j}$  は0又は1であり、 $i=1 \sim 24$ 、 $j=1 \sim 24$  である) を使って次のような符号生成多項式として表す。



13

$$Q_2^i = a_{1,1} D_1 + a_{1,2} D_2 + \cdots + a_{1,24} D_{24}$$

$$I_1^i = a_{2,1} D_1 + a_{2,2} D_2 + \cdots + a_{2,24} D_{24}$$

$$I_2^i = a_{3,1} D_1 + a_{3,2} D_2 + \cdots + a_{3,24} D_{24}$$

$$Q_1^i = a_{4,1} D_1 + a_{4,2} D_2 + \cdots + a_{4,24} D_{24}$$

$$Q_2^i = a_{5,1} D_1 + a_{5,2} D_2 + \cdots + a_{5,24} D_{24}$$

14

(4)

式(1)、(2)を参照すれば、例えば $a_{1,1} = 1$ ,  $a_{1,2} = 1$ ,  $a_{1,3} = 0$ ,  $a_{1,4} = 1$ ,  $a_{1,5} = 1$ ,  $a_{1,6} = 0$ ,  $a_{1,7} = 1$ であり、 $8 \leq j \leq 24$ では $a_{1,j} = 0$ であることは明かである。同様にして他の全ての係数も決定されており、これら全多項式の係数 $a_{i,j}$ を要素とする正方行列Aを図6に示す。即ち、この行列Aは図4又は5に示す符号器の畳み込み符号生成行列を表している。図6から明らかなように、例えば4行9列の係数

$$D_1 = b_{1,1} I_1^i + b_{1,2} I_2^i + b_{1,3} Q_1^i + \cdots + b_{1,24} Q_2^i$$

$$D_2 = b_{2,1} I_1^i + b_{2,2} I_2^i + b_{2,3} Q_1^i + \cdots + b_{2,24} Q_2^i$$

$$D_i = b_{i,1} I_1^i + b_{i,2} I_2^i + b_{i,3} Q_1^i + \cdots + b_{i,24} Q_2^i \quad (5)$$

ここで $b_{i,j}$  ( $i = 1 \sim 24$ ,  $j = 1 \sim 24$ )は正方行列Bの要素であり、“1”又は“0”である。式(5)を復号生成多項式と呼ぶ。

【0025】図6に示す正方行列Aの逆行列は容易に計算することができ、その計算結果を図7に示す。行列Bは原データ復号生成行列を表している。前述の説明から明らかなように符号化率 $R = 3/4$ の場合の畳み込み符号の復号においては4ビットの受信符号データ毎に3ビットの原データを得ることになる。図7の行列Bにおいて、任意の行から開始して連続する3行は何れも図6に示す符号生成行列に従って畳み込み符号化を行う図4又は5に示す符号器によって生成された畳み込み符号から3ビットの原データを決定する3つの復号生成多項式の係数解を与えており、どの連続する3行の係数を使っても復号回路を構成することができる。

【0026】図8は図7においてLaで示す最下3行のグループの係数を使って構成したこの発明による簡易復号回路の実施例を示し、復号すべき符号は前述の図4又は5の符号器によって生成される拘束長 $K = 7$ 、符号化率 $R = 3/4$ の畳み込み符号である。この実施例では24ビットの受信符号データを保持するのに必要な24個のシフト段#1～#24は4行6列に配列され、各行の6つのシフト段は直列に接続され4つのシフトレジスタ $SR_1 \sim SR_4$ が構成され、これら4つのシフトレジスタがシフトレジスタ回路22を構成している。これら4つのシフトレジスタ $SR_1 \sim SR_4$ の入力端子21<sub>1</sub>～21<sub>4</sub>に4ビット毎の入力符号データ $I_1^i, I_2^i, Q_1^i, Q_2^i$ を分配して入力することを繰り返すことにより24個のシフト段の全てがデータが満たされている状

態では第1行第6列(右上角)のシフト段に最も早い、即ち最も古いデータが保持され、第4行第1列(左下角)のシフト段に最も遅れた、即ち最も新しいデータが保持されているので、24個のデータの最も早いデータから最も遅れたデータまでをそれぞれ保持しているシフト段に図に示すように番号#1～#24を順に付けてある。

【0024】

【0027】図8の簡易復号回路には復号結果を3ビットずつ並列出力する3つのmod 2加算器23<sub>1</sub>、23<sub>2</sub>、23<sub>3</sub>が更に設けられている。図7の行グループLa内の第1行目の左から右に並んだ24個の係数は24個のシフト段#1～#24のそれぞれとmod 2加算器23<sub>1</sub>との結合状態を表し、係数“1”のそれぞれ位置に対応するシフト段#1～#4、#6、#9、#15、#17、#19、#20、#23の出力が選択されmod 2加算器23<sub>1</sub>でmod 2加算される。同様に行グループLa内の第2行目の24個の係数は24個のシフト段#1～#24のそれぞれとmod 2加算器23<sub>2</sub>との結合状態を表し、係数“1”の位置に対応するシフト段#5～#9、#11、#12、#15～#19、#21～#23の出力が選択されmod 2加算器23<sub>2</sub>でmod 2加算され、行グループLa内の第3行目の24個の係数は24個のシフト段のそれぞれとmod 2加算器23<sub>3</sub>との結合状態を表し、係数“1”の位置に対応するシフト段#9～#12、#14、#17、#21、#22、#24の出力が選択されmod 2加算器23<sub>3</sub>でmod 2加算される。

【0028】最初の24個の符号データがシフトレジスタ回路22に保持された状態でmod 2加算器23<sub>1</sub>～

23, から得られる出力は復号された原データ  $D_1, D_2, D_3$  である。以降4ビットのデータ  $I_1, I_2, Q_1, Q_2$  がシフトレジスタ回路22にされる毎にデータ  $D_4, D_5, D_6, \dots$  が3ビットずつ出力端子24, 24, 24, に得られる。

【0029】前述のように行グループLaは畳み込み符号の復号を可能とする生成多項式を決める1組の係数解であり、同様に他の任意の連続する3行のグループの何れも同様の畳み込み符号の復号を可能とする生成多項式を決める他の組の係数解となっている。それぞれ連続する3行の係数に従って図8における24個のシフト段#1~#24と3つのmod2加算器23<sub>1</sub>, 23<sub>2</sub>, 23<sub>3</sub>との結線を行えば同様に簡易復号回路を構成することができる。例えばLbで示す最上3行を使って簡易復号回路を構成する場合は、図7から明らかなように行グループLbの右端4つのビット位置の係数は全て“0”となっており、この4つのビット位置に対応するシフト段は3つのmod2加算器23<sub>1</sub>, 23<sub>2</sub>, 23<sub>3</sub>の何れとも結合されない。従ってこの4つのシフト段を省略することができ、その場合シフトレジスタ回路22はシフト段#21~#24が省略された4行5列のシフトレジスタ回路となる。この場合が前述の論文に発表した、この出願の発明者等によって試行錯誤的に構成することができた復号回路に相当する。

【0030】行グループLcは行グループLbにおいて右端4つのビット位置の係数“0”を全て左端に移したものと等価であり、その4つのビット位置に対応するシフト段も同様に省略できるので、行グループLbに従って構成した簡易復号回路と全く同じとなる。この様にK=7, R=3/4の畳み込み符号が4ビット毎に並列にされる簡易復号回路のシフトレジスタ回路22は一般には4行6列のシフトレジスタ回路となるが、図7の場合のように1つまたは複数のシフト段が省略できる係数解の組が得られる場合もある。また図8におけるシフ

$$\begin{aligned} I_1^1 &= D_1 + D_2 + D_4 + D_5 + D_7 \\ I_2^1 &= D_1 + D_3 + D_5 + D_6 + D_8 \\ I_3^1 &= D_3 + D_4 + D_6 + D_7 + D_9 \\ I_4^1 &= D_4 + D_5 + D_7 + D_8 + D_{10} \\ I_5^1 &= D_4 + D_7 + D_9 + D_{10} + D_{12} \\ Q_1^1 &= D_1 + D_4 + D_5 + D_6 + D_7 \\ Q_2^1 &= D_5 + D_6 + D_8 + D_{10} + D_{11} \\ Q_3^1 &= D_7 + D_{10} + D_{11} + D_{12} + D_{13} \end{aligned}$$

次の7ビットの原データ  $D_1 \sim D_{13}$  が符号器に入力されると上式中のデータを全て7ビットシフトして得られる

$$\begin{aligned} I_1^2 &= D_8 + D_9 + D_{11} + D_{12} + D_{14} \\ I_2^2 &= D_9 + D_{10} + D_{12} + D_{13} + D_{15} \\ I_3^2 &= D_{10} + D_{11} + D_{13} + D_{14} + D_{16} \\ I_4^2 &= D_{11} + D_{12} + D_{14} + D_{15} + D_{17} \\ I_5^2 &= D_{12} + D_{14} + D_{16} + D_{17} + D_{18} \\ Q_1^2 &= D_8 + D_{11} + D_{12} + D_{13} + D_{14} \end{aligned}$$

ト段#13のように何れのmod2加算器とも結合されないがデータ間のタイミングを保存するためにのみ使われるものもある。

【0031】図8の実施例では入力符号データが4ビット毎に並列にされる場合の簡易復号回路の例を示したが、図9に示す簡易復号回路の実施例は直列にされる畳み込み符号に対し図8の場合と同様に図7の逆行列で表された係数解の1つの組である行グループLaに従って構成した場合を示す。この実施例ではシフトレジスタ回路22を構成する24個のシフト段#1~#24は全て直列に接続され、これらのシフト段と3つのmod2加算器23<sub>1</sub>, 23<sub>2</sub>, 23<sub>3</sub>が図7の逆行列中の行グループLaの係数に従って図8の場合と同様に接続されている。入力符号データは  $I_1, I_2, Q_1, Q_2$  の順に繰り返しシフトレジスタ回路22にされ、全シフト段#1~#24にデータが初めて満たされた状態でmod2加算器23<sub>1</sub>, 23<sub>2</sub>, 23<sub>3</sub>の出力が復号された原データ  $D_1, D_2, D_3$  としてラッチ25<sub>1</sub>, 25<sub>2</sub>, 25<sub>3</sub>に取り込まれ、出力端子24<sub>1</sub>, 24<sub>2</sub>, 24<sub>3</sub>に出力される。以降は4ビットの符号データ  $I_1, I_2, Q_1, Q_2$  がされる毎にmod2加算器23<sub>1</sub>, 23<sub>2</sub>, 23<sub>3</sub>から3ビットの復号データがラッチ25<sub>1</sub>, 25<sub>2</sub>, 25<sub>3</sub>に取り込まれる。

【0032】上述では拘束長K=7、符号化率R=n/m=3/4の畳み込み符号に対する簡易復号回路の例を説明したが、同じ拘束長K=7、で例えば符号化率R=n/m=7/8の畳み込み符号の場合についてその簡易復号回路を構成する手順を簡単に説明する。最初の13ビットの原データ  $D_1 \sim D_{13}$  を用いて次の符号生成多項式に従った符号器により最初の8ビットの畳み込み符号データ  $I_1, I_2, I_3, I_4, I_5, Q_1, Q_2, Q_3$  を生成するものとする。

【0033】

(6)

次式により次の8ビット1組の符号データが得られる。

【0034】

$$Q_i' = D_{i,1} + D_{i,2} + D_{i,3} + D_{i,4} + D_{i,5}$$

$$Q_i' = D_{i,4} + D_{i,5} + D_{i,6} + D_{i,7} + D_{i,8}$$

( 7 )

更に原データDを7ビット符号器に入力する毎に8ビット1組の符号データが増えるので、8ビット符号データの生成をx回繰り返したとき、得られる符号生成多項式の数(全符号データ数)がそれまでに入力された原データDの数と等しいかそれより大きいとすると、 $8x \geq 13 + 7(x-1)$  が成り立つから最小のxの値は $x=6$ となる。この時得られる $8x=48$ 個の符号データをそれぞれ表す符号生成多項式の各係数 $a_{i,j}$  ( $i=1 \sim 48$ ,  $j=1 \sim 48$ ) で表すと、式(6)、(7)からこれらの係数 $a_{i,j}$  は図10に示す正方行列Aで表すことができる。これら48個の符号生成多項式の係数がこのように決められているので、48個の符号化データが与えられれば逆に48個の原データを決定することができ、それら原データを生成する復号生成多項式の係数 $b_{i,j}$  ( $i=1 \sim 48$ ,  $j=1 \sim 48$ ) は図10に示す正方行列Aの逆行列 $B=A^{-1}$  によって与えられ、その逆行列Bの計算結果を図11に示す。従って、簡易復号回路においては $8x=48$ ビットの入力畳み込み符号データを保持するシフト段数を有するシフトレジスタ回路を設ければ、新しく入力された8ビットの符号データとそれ以前の40ビットの符号データから図11で決められた

$$mx \geq K + n - 1 + n(x-1)$$

を満足する必要がある。従って列数(例えば図8の実施例の各シフトレジスタ $SR_1 \sim SR_4$ のシフト段数)x

$$x \geq (K-1) / (m-n)$$

を満足する整数を選べばよく、またシフトレジスタ回路

$$N = xm \geq m(K-1) / (m-n)$$

で表される。簡易復号回路の構成を小さくするにはxの

$$x = [(K-1) / (m-n)]$$

によってxの値を決定すればよい。但し記号[p]は実数p以上の最小の整数を表すものとする。あるいは上式(9)、(10)において $(K-1) / (m-n)$  が整数とならない場合は、 $(K-1) / (m'-n)$  が整数となるmを越えない最大の整数 $m'$ を求めてmの代わりに用いてもよい。一般に $m' = n+1$ に選べば式(9)のxの値は拘束長Kが如何なる値でも常に整数となる。即ち、例えば式(1)のようなm個の畳み込み符号生成多項式から $m' = n+1$ 個を選択して前述と同様にnビットのデータを追加する毎に $m'$ 個の生成多項式を増やしていく手順で $xm'$ 個の生成多項式を得て、それら多項式の係数を要素とする $m \times m \times x$ の正方行列Aを求める。この場合、シフトレジスタ回路22内の1行のシフト段数xは $K-1$ となる。何れの方法においても得られた符号生成行列Aの行列式が0となった場合は逆行列 $B=A^{-1}$ を求めることはできない。その時は式(9)を満足する他のxの値を選べば逆行列を求められる可能性がある。

【0037】この様にこの発明によれば従来構成できな

係数の7つの復号生成多項式を使って7ビットの原データを復号する事ができる。

【0035】図11の逆行列において、任意の行から開始する連続する7行は何れも復号生成多項式の係数解の組を与える。例えば行グループLaで示す最上7行の係数に従って構成した簡易復号回路を図12に示す。図8の場合と同様に行グループLa内の第1~7行の係数

“1”の位置はシフト段#1~#48中の、mod 2加算器23<sub>1</sub>~23<sub>7</sub>と結合すべきシフト段番号を表している。この簡易復号回路により、最初の48ビットの符号データがシフト段#1~#48に保持された状態で原データD<sub>1</sub>~D<sub>7</sub>がmod 2加算器23<sub>1</sub>~23<sub>7</sub>から復号されて出力され、以降入力符号データが8ビット与えられる毎に7ビットの原データが復号される。図11の簡易復号回路も図9と同様に全てのシフト段が直列接続された構成に変形できることは明かである。

【0036】一般に拘束長K、符号化率 $R=n/m$ の畳み込み符号に対する簡易復号回路において、シフトレジスタ回路22に保持しなければならない全データビット数 $N=mx$ は式(3)と同様に次式

( 8 )

は次式

( 9 )

22内の必要な全シフト段数Nは次式

( 10 )

値を小さく選ぶことが好ましく、その場合は次式

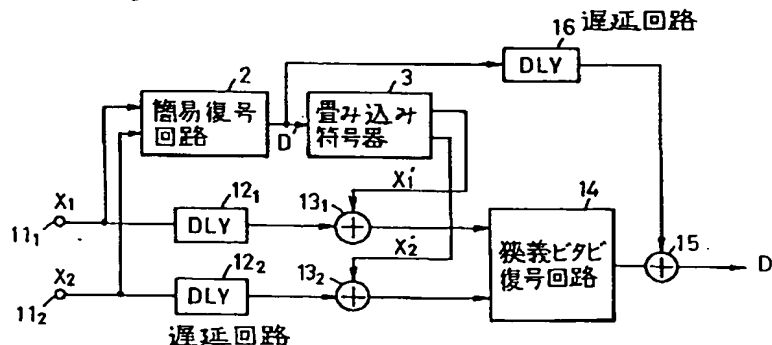
( 11 )

かった任意の符号化率 $n/m$ の復号回路を例えば図8、9、12に示すようにシフトレジスタ回路と論理回路により簡単に実現できる。この発明の適用例として、図1と同様に簡易復号回路を用いて構成した符号化率 $n/m$ のSST型ビタビ復号器の構成を図13に示す。伝送路上で誤りの加わったmビットの受信畳み込み符号データ(軟判定データ) $X_1 \sim X_m$ から符号化率 $n/m$ のこの発明の簡易復号回路20を用いてnビットの原データDの推測を行う。次にnビットの推測原データD'を送信側畳み込み符号器と同じ構成の符号器3で再符号化し、この再符号化畳み込み符号と受信畳み込み符号 $X_1 \sim X_m$ を比較器(排他的論理和回路)13<sub>1</sub>~13<sub>m</sub>で比較して検出される伝送路誤りに起因したビットパターン異常を狭義のビタビ復号回路14に入力し、対応する推定データD'の誤りを復号する。この結果得られた推定データD'の誤りと簡易復号回路20の出力である推定原データD'をmod 2加算器15<sub>1</sub>~15<sub>m</sub>でmod 2加算することにより推定原データD'中の誤りを訂正してnビットの最終復号データDを得る。

【0039】

【图 1】

FIG. 1



【図面の簡単な説明】

【図3】図2の符号器に対応する簡易復号回路の構成図。

【図5】図4の符号器を変形した非パンクチュアド型符号器の例を示す図。

【図6】図4の符号器の符号生成多項式の係数を要素とする正方行列を示す図。

【図7】 図6の正方行列の逆行列を示す図。

【図8】図7により求められた復号生成多項式の係数に従って構成した簡易復号回路の実施例を示す図。

【図9】 図8の復号回路の変形実施例を示す図。

【図10】符号化率 $R=7/8$ 、拘束長 $K=7$ の符号生成多項式の係数を要素とする正方形行列を示す図。

【図 1 1】 図 1 0 の行列の逆行列を示す図。

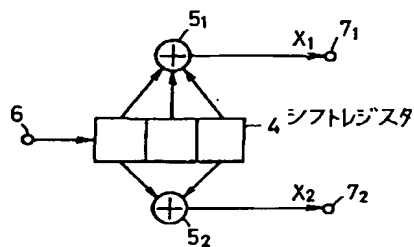
【図12】図11により求められた復号生成多項式の係数に従って構成した簡易復号回路の実施例を示す図。

【図１３】この発明の簡易復号回路を適用したＳＳＴ型  
ビタビ復号器の構成を示すブロック図。

【図１４】この発明の簡易復号回路を適用した伝送路符号誤り測定装置の構成を示すブロック図。

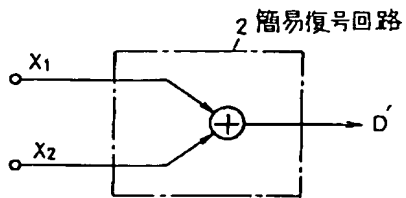
【図 2】

FIG. 2



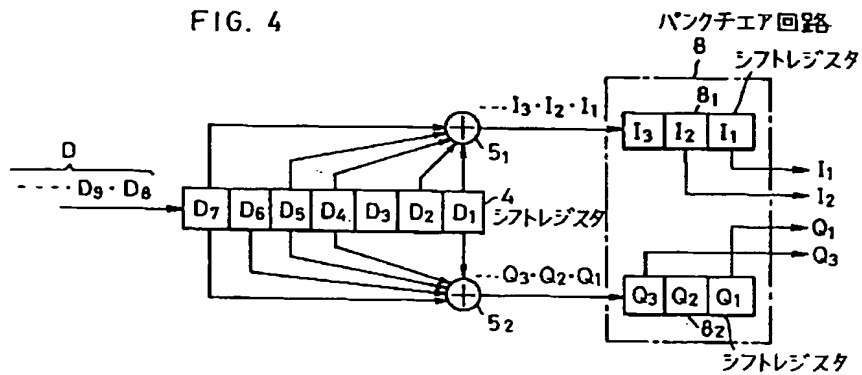
【図 3】

FIG. 3



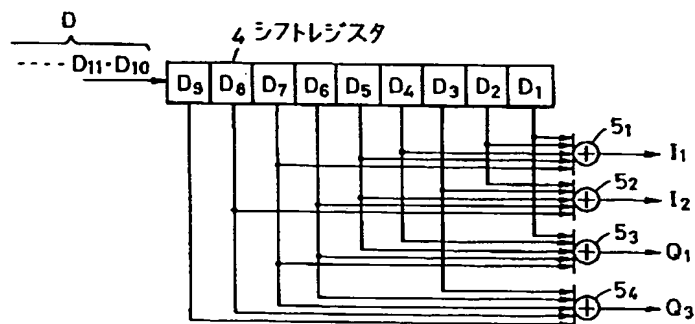
【図 4】

FIG. 4



【図 5】

FIG. 5



【図 6】

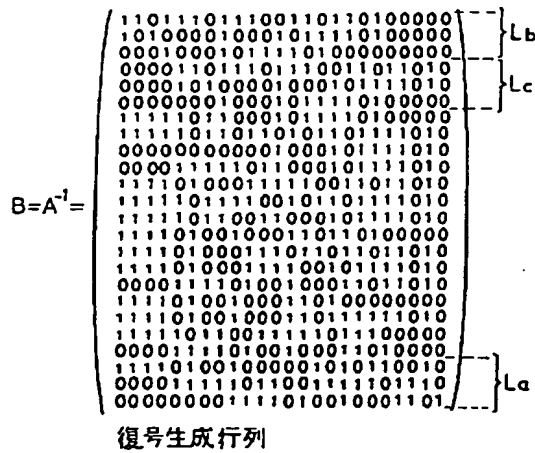
FIG. 6

$$A = \begin{pmatrix} 110110100000000000000000 \\ 011011010000000000000000 \\ 100111100000000000000000 \\ 001001110000000000000000 \\ 000110110100000000000000 \\ 000011011010000000000000 \\ 000100111100000000000000 \\ 000001001110000000000000 \\ 000000110110100000000000 \\ 000000011011010000000000 \\ 000000001001110000000000 \\ 000000000110110100000000 \\ 000000000010011100000000 \\ 000000000001001110000000 \\ 000000000000100111000000 \\ 000000000000010011100000 \\ 000000000000001001110000 \\ 000000000000000100111000 \\ 000000000000000010011100 \\ 000000000000000001001101 \\ 000000000000000000100110 \\ 000000000000000000010011 \\ 00000000000000000000100111 \end{pmatrix}$$

符号生成行列

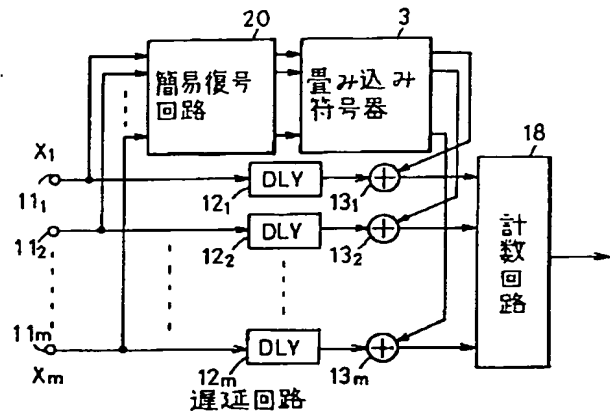
【図 7】

FIG. 7



【図 1 4】

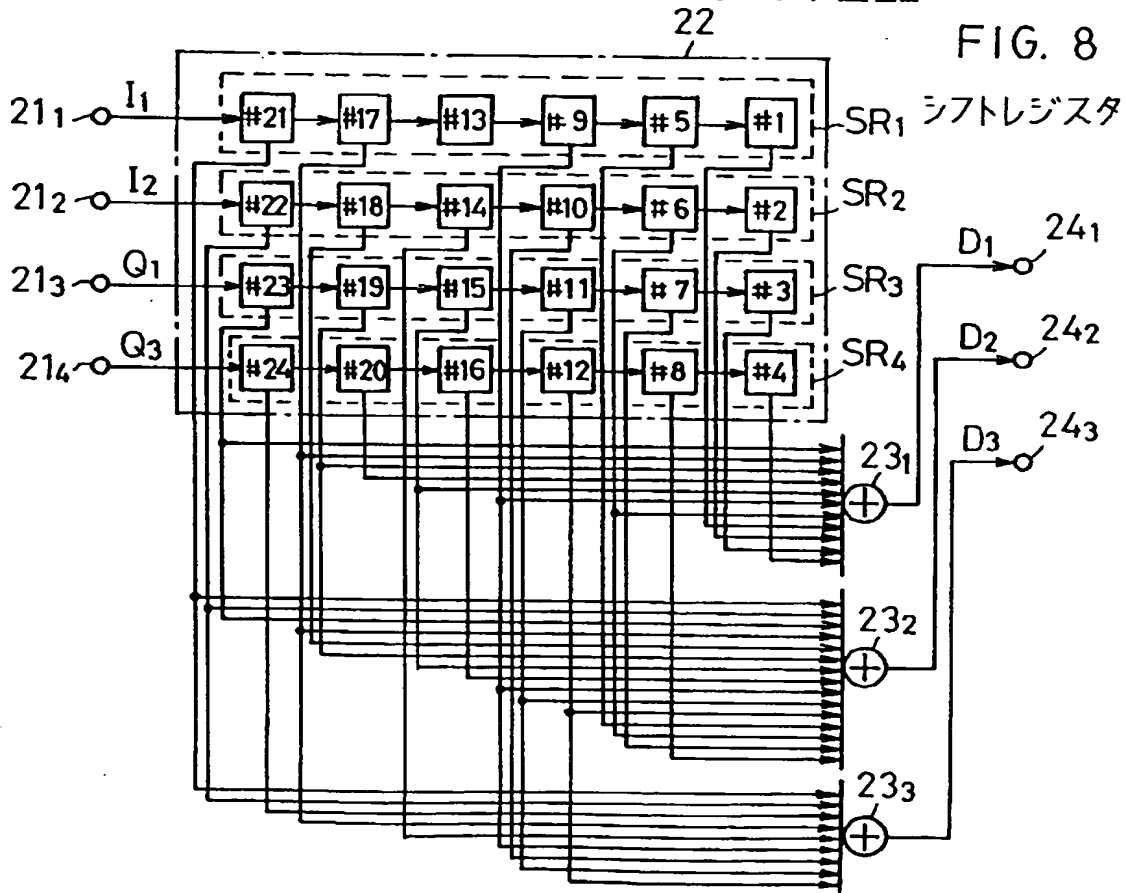
FIG. 14



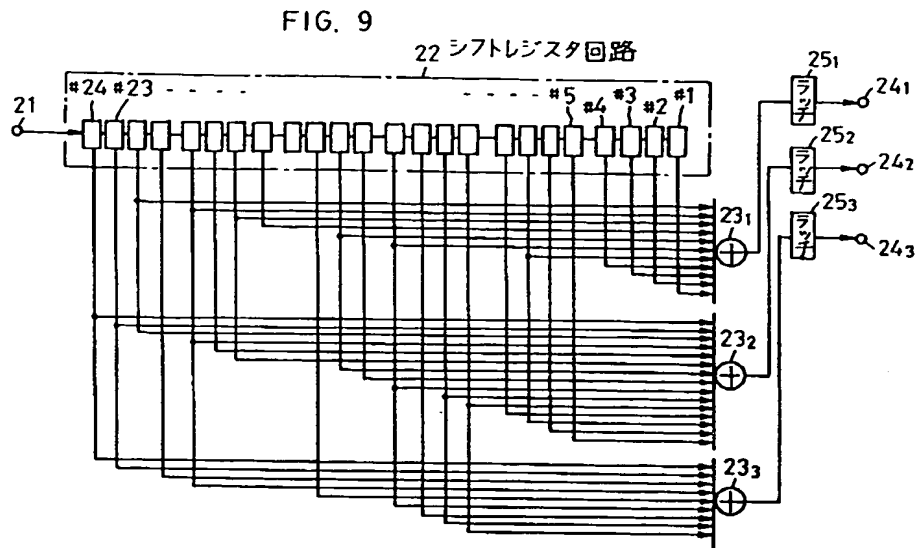
【図 8】

シフトレジスタ回路

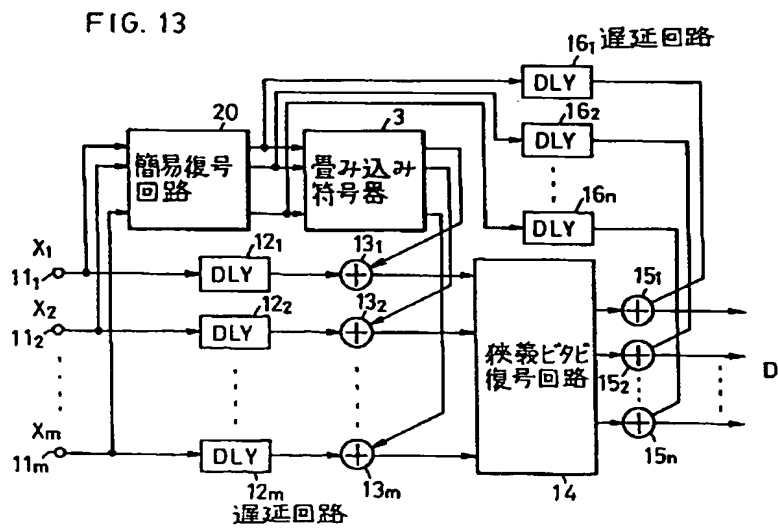
FIG. 8



【図 9】



【図 13】









【図 12】

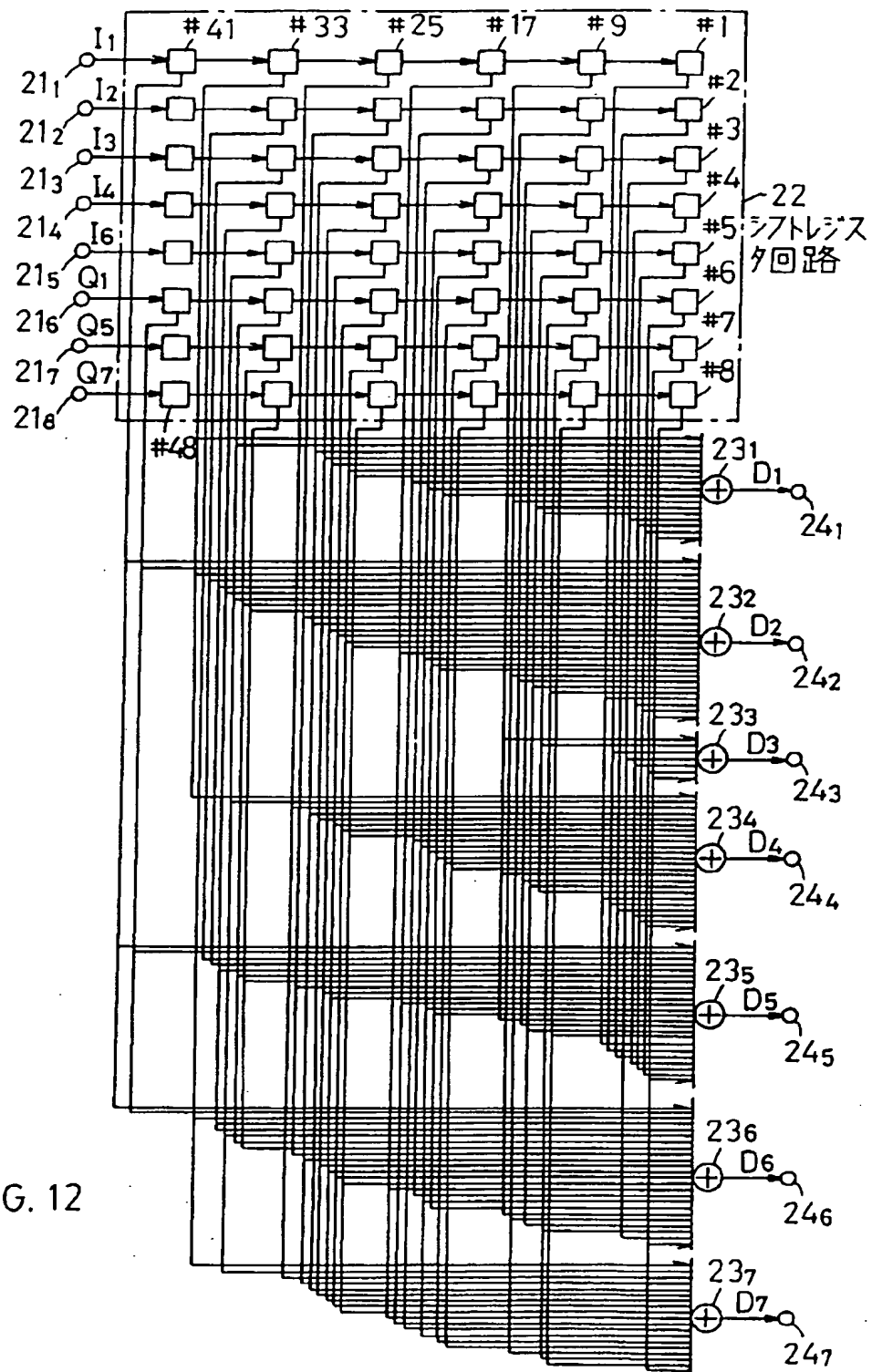


FIG. 12

フロントページの続き

(72)発明者 加藤 修三  
東京都千代田区内幸町 1 丁目 1 番 6 号 日  
本電信電話株式会社内